

Due Date: August 29, 2007

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re Application of:)	
)	
Inventor: Kenneth A. Vadella et al.)	Examiner: Charles E. Anya
)	
Serial #: 10/607,119)	Group Art Unit: 2194
)	
Filed: June 26, 2003)	Appeal No.: _____
)	
Title: COMMUNICATION MECHANISM)	
BETWEEN DISCONNECTED APPLICATIONS)	
<u>IN A WEB BROWSER</u>)	

BRIEF OF APPELLANTS

MAIL STOP APPEAL BRIEF - PATENTS
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

In accordance with 37 CFR §41.37, Appellants hereby submit the Appellants' Brief on Appeal from the final rejection in the above-identified application, as set forth in the Office Actions dated March 29, 2007 and June 18, 2007.

Please charge the amount of \$500 to cover the required fee for filing this Appeal Brief as set forth under 37 CFR §41.37(a)(2) and 37 CFR §41.20(b)(2) to Deposit Account No. 50-0494 of Gates & Cooper LLP. Also, please charge any additional fees or credit any overpayments to Deposit Account No. 50-0494.

I. REAL PARTY IN INTEREST

The real party in interest is Autodesk, Inc., the assignee of the present application.

II. RELATED APPEALS AND INTERFERENCES

There are no related appeals or interferences for the above-referenced patent application.

III. STATUS OF CLAIMS

Claims 1-24 are pending in the application.

Claims 1-24 stand rejected.

Appellants are appealing the rejection of claims 1-24.

IV. STATUS OF AMENDMENTS

No amendments to the claims have been made subsequent to the final Office Action.

V. SUMMARY OF CLAIMED SUBJECT MATTER

Briefly, Appellants' invention, as recited in independent claims 1, 7, and 13, is generally directed to an invention that performs enabling communication between disconnected applications (see paragraph [0003]-page 2, lines 11-14). More specifically, disconnected applications as used in the present invention and as explicitly claimed therein provide that a disconnected application is unaware of the secondary application (see paragraph [0007]-page 3, line 23-page 4, line 6; paragraph [0029]-page 10, lines 2-17; paragraph [0047]-page 15, lines 19-22; FIGs. 2, 3, 4, and 5). In this regard, the disconnected applications of the present invention are applications that do not know anything about each other. One application (referred to in the claims as a secondary application) creates a bridge object (FIGs. 4 and 5; paragraphs [0044]-[0045]-page 14, line 22-page 15, line 9). Such a bridge object is not part of either application and allows the applications to communicate with each other through an interface (paragraph [0049]-page 14, lines 10-15; FIGs. 4 and 5). In this regard, the claims explicitly provide that an interface for the bridge object enables communication with the secondary application through the bridge object (paragraph [0049]-page 14, lines 10-15; FIGs. 4 and 5). The interface for the

bridge object is registered in a global interface table (GIT) and a cookie is retrieved (from the GIT) in response (paragraph [0048]-page 15, line 23-page 16, line 9; paragraph [0050]-page 16, line 16-page 17, line 3; paragraph [0053]; page 17, lines 14-21; FIGs. 4 and 5). Such a cookie comprises information for utilizing the interface for the bridge object. The claims then explicitly provide for storing the cookie in a location that is accessible to the disconnected application such that the cookie can be retrieved to enable use of the interface (paragraph [0048]-page 15, line 23-page 16, line 9; paragraph [0050]-page 16, line 16-page 17, line 3; paragraph [0053]; page 17, lines 14-21; FIGs. 4 and 5).

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Whether claims 1-24 are unpatentable under 35 U.S.C. §103(a) as being rendered obvious over Platform SDK: COM IGlobalInterface Table (IGlobalInterfaceTable) pages 1-2 in view of U.S. Pub. No. 2004/0205734 to Srinivasan et al (Srinivasan). Such ground is presented for review herein.

VII. ARGUMENT

A. Claims 1-24 Are Patentable Under 35 U.S.C. §103(a) over Platform SDK: COM IGlobalInterface Table (IGlobalInterfaceTable) pages 1-2 in view of U.S. Pub. No. 2004/0205734 to Srinivasan et al (Srinivasan).

1. Independent Claims 1, 7, and 13

In view of the above-described limitations, there are several unique, novel, and nonobvious aspects of the invention. Such aspects include the storage of the cookie in any globally accessible location. The prior art fails to teach, disclose, or suggest the use or storage of the cookie whatsoever. A second aspect includes that the applications are disconnected/independent and are unaware of each other – yet can communicate via the bridge object. A third aspect is that the interface bridge object is not part of either application. A fourth aspect is that the interface placed in the GIT is from the bridge object, rather than from either application. Again, the applications have no direct connection to each other and are disconnected (yet can communicate via the bridge object and the information contained in the

GIT).

On pages (2)-(6) of the Office Action, claims 1-24 were rejected under 35 U.S.C. §103(a) as being obvious in view of the combination of Platform SDK: COM IGlobalInterfaceTable (IGlobalInterfaceTable) and Srinivasan et al., U.S. Publication 2004/020734 (Srinivasan).

Specifically, claim 1, 7 and 13 was rejected as follows:

As to claim 1, IGlobalInterfaceTable teaches a computer-implemented method for enabling communication between applications (“...any apartment...any other apartment...” page 1 line 3), comprising: creating a bridge object in a secondary application (“...an object...” page 1 line 1), wherein an interface for the bridge object enables communication with the secondary application through the bridge object (“...an interface...” page 1 line 1); registering the interface for the bridge object in a global interface table (GIT) (“Register...” page 1 lines 5/37-38, “...register...” page 2 line 5); retrieving a cookie from the GIT in response to the registration, wherein the cookie comprises information for utilizing the interface for the bridge object (“...a cookie...” page 2 line 6, “...get a cookie...” page 2 line 5); and storing the cookie in an environment variable, wherein the environment variable is accessible to a application such that the cookie may be retrieved to enable use of the interface (“...GetInterfaceFromGlobal method...this cookie...” page 1 lines 39-41).

IGlobalInterfaceTable is silent with reference to disconnected applications. Tock teaches disconnected applications (“...offline...” page 1 paragraph 0007, “...disconnected state...” page 9 paragraph 0096).

Srinivasan teaches disconnected applications (Active X Components 135 page 1 paragraph 0008) and the disconnected application is unaware of the secondary application (“...cannot directly call...” page 1 paragraphs 0008/0011).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Tock and IGlobalInterfaceTable because the teaching of Tock would improve the system of IGlobalInterfaceTable by providing a method for allowing a client application to operate offline from a server (Tock page 1 paragraph 0007).

As to claims 7 and 13, see the rejection of claim 1 above.

Appellants traverse the above rejections for one or more of the following reasons:

- (1) Neither IGlobalInterfaceTable nor Srinivasan teach, disclose or suggest the storage of the cookie in a location that is accessible to a disconnected application;
- (2) Neither IGlobalInterfaceTable nor Srinivasan teach, disclose or suggest a disconnected application that is unaware of the secondary application; and
- (3) Neither IGlobalInterfaceTable nor Srinivasan teach, disclose or suggest a secondary application and disconnected application executing within the same process but in different apartments.

In rejecting the claims, the Office Action primarily relies on the IGlobalInterfaceTable reference. Specifically, with respect to the storage of the cookie in a location accessible to the disconnected application (e.g., an environment variable), the Office Action refers to page 1, lines 39-41. Page 1, lines 37-41 provides:

After calling the CoCreateInstance function, register the interface you want to make available processwide from the apartment in which it resides with a call to the RegisterInterfaceInGlobal method. This supplies a pointer to a "cookie" (through the *pdwCookie* parameter) that identifies the interface and its location. An apartment that wants a pointer to this interface then calls the GetInterfaceFromGlobal method with this cookie, and the implementation then supplies an interface pointer to the calling apartment. To revoke the interface's global registration, any apartment may call the RevokeInterfaceFromGlobal method.

As can be seen from this text (and the remainder of IGlobalInterfaceTable), there is no description, explicit or implicit, regarding the storage or what to do with the cookie. Instead, the reference merely describes the supplying of a pointer to a cookie that identifies an interface and its location. A method is then called with the cookie and an interface is supplied to a calling apartment. In this regard, the cookie of IGlobalInterfaceTable could merely be passing around the cookie as an argument to various functions. The present claims are unique in that the cookie of the claimed invention is stored in a location that is accessible to both the disconnected application and the secondary application. For example, the cookie could be stored in a globally accessible location such as a database, file system, or registry. The dependent claims explicitly provide that the location comprises an environment variable. Nowhere in IGlobalInterfaceTable is there any description, suggestion, or remote reference to the storing of the cookie in any location, not to mention the storage in an environment variable as claimed.

In response to the above, arguments, the final Office Action essentially repeats the rejections. Appellants again reassert that above and note that lines 39-41 of IGlobalInterfaceTable provide for supplying a pointer to a cookie that identifies the interface and its location. The "its location" modifies the term interface and describes the location of the interface. In this regard, the location described in IGlobalInterfaceTable does not refer to the location of the cookie itself. Such a location of a cookie is not contemplated in IGlobalInterfaceTable.

In view of the above, Appellants note that the present application addressed security issues that needed to be addresses because of the use of a web browser. To overcome the security limitations imposed by such a web browser, the particular use and method of the cookie and bridge were developed (and is currently set forth in the claims). Such a methodology did not exist in the past. In this regard, the IGlobalInterfaceTable reference clearly falls within the prior art and does not provide a method for accessing the bridge object or storing a cookie (containing information for such a bridge object) in a location that is accessible to a disconnected application. Again, IGlobalInterfaceTable completely fails to teach, disclose, or suggest, explicitly or implicitly, any storage of a cookie in a globally accessible location. In addition, the use and manner of use of the cookie is neither taught not disclosed in IGlobalInterfaceTable.

The Office Action continues and submits that IGlobalInterfaceTable is silent with respect to disconnected applications and the disconnected application being unaware of the secondary application. Instead, the Office Action relies on Srinivasan. Appellants respectfully traverse such an assertion. As set forth in the claim limitations, the disconnected applications of the present invention do not refer to applications that are merely unable to directly call each other. Instead, the claim limitations explicitly provide that the disconnected application is unaware of the secondary application. Srinivasan paragraph [0007] serves to actually teach away from such a limitation. In this regard, paragraph [0007] describes a COM client looking for a calculator through a Jini brokering service. In this regard, the COM client is explicitly aware of a calculator. In fact, the COM client searches by specifying GUIDs on behalf of a client. The Jini broker finds the desired ActiveX component and returns Java objects. Thus, contrary to that asserted in the final Office Action, the applications are clearly aware of each other.

Further, rather than utilizing a cookie to retrieve information for utilizing an interface for a bridge object, the Jini application merely wraps serialized object code as an ActiveX Java service so that it can be accessed by a COM application (see paragraph [0008]). Such a use is not even remotely relevant to the present claims. In this regard, wrapping up an object so that a COM application can use a Java object is not similar in any way, shape, or form, to the explicit and specific limitations set forth in the present application.

In response to the above asserted arguments, the Advisory Action merely states that the Examiner maintains that the final rejection of 3/29/07 covers the invention as claimed.

In view of the above, Appellants respectfully request reversal of the rejections.

2. Dependent Claims 2, 8, and 14

Dependent claims 2, 8, and 14 provide that the secondary application is a project hosting environment. In rejecting these claims, the final Office Action relies on Srinivasan (application 110, page 1, paragraphs 0007-0010).

Appellants note that Srinivasan's Jini application is a stand alone application rather than an application executing in a web browser or a project hosting environment. Nowhere in Srinivasan is there even a remote reference to a project or of a host. In this regard, separate electronic searches of Srinivasan for the terms "project" and "host" provides no results whatsoever. Without even mentioning the word "project" or the word "host", Srinivasan cannot possibly disclose or render obvious a hosting environment for a project.

In view of the above, Appellants respectfully request reversal of the rejections.

3. Dependent Claims 3, 9, and 15 are Not Separately Argued

4. Dependent Claims 4, 10, and 16

Dependent claims 4, 10, and 16 provide that the registering of the interface for the bridge object places a pointer to the interface for the bridge object in the global interface table. In rejecting these claims, the final Office Action merely refers to the interface pointer on page 1 lines 8-9 of the IGlobalInterfaceTable reference.

As stated above, one of the unique elements of the invention is that the bridge object is not part of either application and the interface is from the bridge object rather than from either application. Dependent claims 4, 10, and 16 provide further limitations in this regard and specify that the registering of the interface places a pointer to the interface FOR THE BRIDGE OBJECT in the global interface table. The cited portion of IGlobalInterfaceTable merely supplies a pointer to a cookie that identifies the interface and its location. Such a pointer to a cookie is not

a pointer to an interface as claimed. In this regard, a cookie is not an interface. As claimed, the cookie comprises information for utilizing the interface. Appellants also direct the attention of the Board to claims 5, 11, and 17 discussed below.

In view of the above, Appellants respectfully request reversal of the rejections.

5. Dependent Claims 5, 11, and 17

Dependent claims 5, 11, and 17 depend on claims 4, 10, and 16 and provide that the cookie identifies the pointer and a location of the interface. In rejecting these claims, the final Office Action merely recites the term “identifies” on page 1, line 39 of IGlobalInterfaceTable.

When viewing these claims in combination with claims 4, 10, and 16, Appellants note that it would be impossible to accept the Examiner’s arguments with respect to claims 4, 10, and 16. In this regard, IGlobalInterfaceTable provides a pointer to a cookie while these claims provides that the cookie identifies the pointer to the interface. Such a contradiction cannot be rationalized. If IGlobalInterfaceTable provides a pointer to a cookie, then the same language cannot be used to indicate that the cookie identifies the pointer and location of the interface.

In view of the above, Appellants respectfully request reversal of the rejections.

6. Dependent Claims 6, 12, and 18

Dependent claims 6, 12, and 18 provide for a disconnected application: extracting the cookie from the location, accessing the cookie to enable use of the interface for the bridge object, and communicating with the secondary application using the interface for the bridge object.

In rejecting these claims, the final Office Action recites the “GetInterfaceFromGlobal method” on page 1, lines 40-41 and the disconnected application from Srinivasan (ActiveX Component 135 page 1 paragraph 0008).

Appellants respectfully disagree with and traverse the rejections. As set forth in IGlobalInterfaceTable, the GetInterfaceFromGlobal method is called with the cookie which supplies an interface pointer to a calling apartment. However, such a description fails to teach, describe, or suggest the extracting of a cookie from a location that is accessible to a disconnected application as claimed. Further, as stated above, Srinivasan’s ActiveX component is not

equivalent to a disconnected application as claimed. Again, the claimed disconnected application is unaware of the secondary application while Srinivasan's components is fully aware and actually teaches away from such a lack of knowledge as claimed.

In view of the above, Appellants respectfully request reversal of the rejections.

7. Dependent Claims 19, 21, and 23

Dependent claims 19, 21, and 23 provide for storing the cookie in an environment variable.

In rejecting these claims, the final Office Action merely recites IGlobalInterfaceTable page 1, lines 38-41. Such a description does not even remotely reference an environment variable. In fact, Appellants submit that a use of an environment variable is not even contemplated or mentioned anywhere in IGlobalInterfaceTable. Again, the claims provide for explicit claim limitations. Under MPEP §2142 and 2143.03 "To establish *prima facie* obviousness of a claimed invention, all the claim limitations must be taught or suggested by the prior art. *In re Royka*, 490 F.2d 981, 180 USPQ 580 (CCPA 1974). "All words in a claim must be considered in judging the patentability of that claim against the prior art." *In re Wilson*, 424 F.2d 1382, 1385, 165 USPQ 494, 496 (CCPA 1970)." The Office Action cannot merely ignore the claim limitations directed towards environment variables and recite a location of an interface, which has no relevance with respect to the storage of a cookie.

Instead of teaching such an environment variable, IGlobalInterfaceTable merely teaches a pointer to a "cookie" that identifies an interface and its location. Such a teaching does not contemplate, explicitly or implicitly, an environment variable nor the storage of the cookie in such an environment variable.

In view of the above, Appellants respectfully request reversal of the rejections.

8. Dependent Claims 20, 22, and 24

Dependent claims 20, 22, and 24 provide that the secondary application and the disconnected application are executing within a same process but in different apartments. Thus,

as claimed, while both applications are in the same process, they are unaware of each other. Such a capability is wholly and completely lacking from IGlobalInterfaceTable.

In view of the above, Appellants respectfully request reversal of the rejections.

B. Conclusion

In light of the above arguments, Appellants respectfully submit that the cited references do not anticipate nor render obvious the claimed invention. More specifically, Appellants' claims recite novel physical features which patentably distinguish over any and all references under 35 U.S.C. §§ 102 and 103. As a result, a decision by the Board of Patent Appeals and Interferences reversing the Examiner and directing allowance of the pending claims in the subject application is respectfully solicited.

Respectfully submitted,

GATES & COOPER LLP

Attorneys for Appellant(s)

Howard Hughes Center
6701 Center Drive West, Suite 1050
Los Angeles, California 90045
(310) 641-8797

Date: August 28, 2007

By: /Jason S. Feldmar/
Name: Jason S. Feldmar
Reg. No.: 39,187

JSF/

G&C 30566.243-US-U1

CLAIMS APPENDIX

1. A computer-implemented method for enabling communication between disconnected applications, comprising:
 - a secondary application creating a bridge object, wherein an interface for the bridge object enables communication with the secondary application through the bridge object;
 - registering the interface for the bridge object in a global interface table (GIT);
 - retrieving a cookie from the GIT in response to the registration, wherein the cookie comprises information for utilizing the interface for the bridge object; and
 - storing the cookie in a location that is accessible to a disconnected application such that the cookie may be retrieved to enable use of the interface, and wherein the disconnected application is unaware of the secondary application.
2. The method of claim 1, wherein the secondary application comprises a project hosting environment.
3. The method of claim 1, wherein the disconnected application comprises an ActiveX control.
4. The method of claim 1, wherein the registering of the interface for the bridge object in the GIT comprises placing a pointer to the interface for the bridge object in the GIT.
5. The method of claim 4, wherein the cookie identifies the pointer and a location of the interface.
6. The method of claim 1, further comprising:
 - the disconnected application extracting the cookie from the location;
 - the disconnected application accessing the cookie to enable use of the interface for the bridge object; and

the disconnected application communicating with the secondary application using the interface for the bridge object.

7. An apparatus for enabling communication between disconnected applications in a computer system comprising:

- (a) a computer system having a memory and a data storage device coupled thereto;
- (b) a secondary application performed by the computer;
- (c) a bridge object created by the secondary application, wherein an interface for the bridge object enables communication with the secondary application through the bridge object;
- (d) a global interface table (GIT) configured to:
 - (i) accept registration of the interface for the bridge object;
 - (ii) return a cookie in response to the registration, wherein the cookie comprises information for utilizing the interface for the bridge object; and
- (e) a location configured to store the cookie, wherein the location is accessible to a disconnected application such that the cookie may be retrieved to enable use of the interface.

8. The apparatus of claim 7, wherein the secondary application comprises a project hosting environment.

9. The apparatus of claim 7, wherein the disconnected application comprises an ActiveX control.

10. The apparatus of claim 7, wherein the GIT accepts the registration of the interface for the bridge object by storing a pointer to the interface for the bridge object.

11. The apparatus of claim 10, wherein the cookie identifies the pointer and a location of the interface.

12. The apparatus of claim 7, wherein the disconnected application is configured to:

extract the cookie from the location;
access the cookie to enable use of the interface for the bridge object; and
communicate with the secondary application using the interface for the bridge object.

13. An article of manufacture comprising a program storage medium readable by a computer and embodying one or more instructions executable by the computer to perform a method for enabling communication between disconnected applications in a computer system, the method comprising:

a secondary application creating a bridge object, wherein an interface for the bridge object enables communication with the secondary application through the bridge object;

registering the interface for the bridge object in a global interface table (GIT);

retrieving a cookie from the GIT in response to the registration, wherein the cookie comprises information for utilizing the interface for the bridge object; and

storing the cookie in a location that is accessible to a disconnected application such that the cookie may be retrieved to enable use of the interface.

14. The article of manufacture of claim 13, wherein the secondary application comprises a project hosting environment.

15. The article of manufacture of claim 13, wherein the disconnected application comprises an ActiveX control.

16. The article of manufacture of claim 13, wherein the registering of the interface for the bridge object in the GIT comprises placing a pointer to the interface for the bridge object in the GIT.

17. The article of manufacture of claim 16, wherein the cookie identifies the pointer and a location of the interface.

18. The article of manufacture of claim 13, wherein the method further comprises:
the disconnected application the cookie from the location;
the disconnected application accessing the cookie to enable use of the interface for the bridge object; and
the disconnected application communicating with the secondary application using the interface for the bridge object.
19. The method of claim 1, wherein the location comprises an environment variable.
20. The method of claim 1, wherein the secondary application and disconnected application are executing within a same process but in different apartments.
21. The apparatus of claim 7, wherein the location comprises an environment variable.
22. The apparatus of claim 7, wherein the secondary application and disconnected application are executing within a same process but in different apartments.
23. The article of manufacture of claim 16, wherein the location comprises an environment variable.
24. The article of manufacture of claim 16, wherein the secondary application and disconnected application are executing within a same process but in different apartments.

EVIDENCE APPENDIX

None.

RELATED PROCEEDINGS APPENDIX

None.